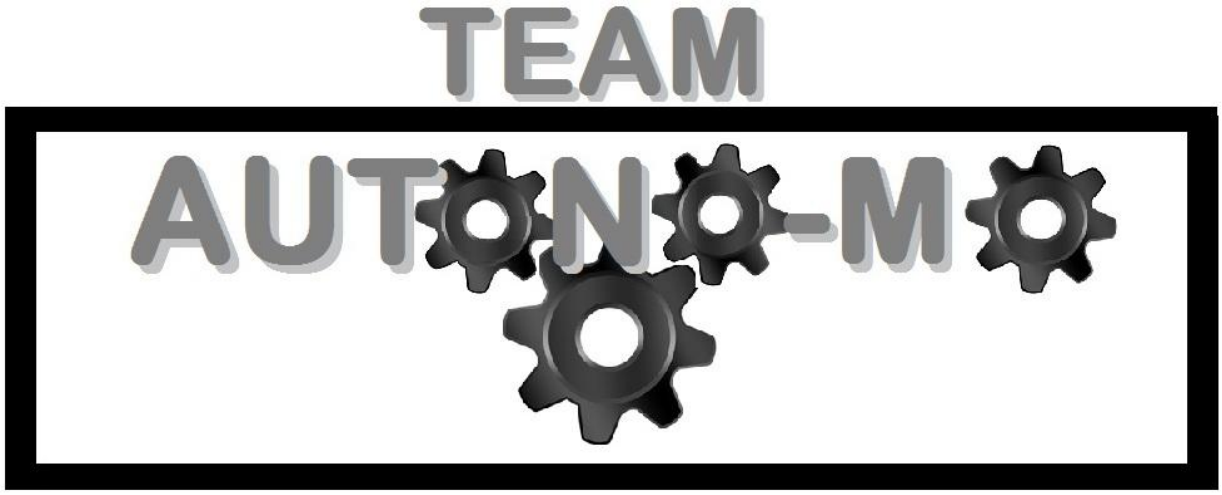
Department of Computer Science and Engineering
The University of Texas at Arlington



Team: Autono-Mo

Project: Jacobia

Team Members:
Bill Butts
Yunesh Shakya
Lance Storey
Darius Salemizadeh

Last Updated: April 21, 2012 – 3:00 PM

Table of Contents

Team: Autono-Mo i

Table of Contents ii

Document Revision History vi

List of Figures vii

List of Tables viii

1. Product Concept 1

1.1 Purpose and Use 1

1.2 Intended Audience 1

2. Product Description and Functional Overview 3

2.1 Features and Functions 3

2.2 External Inputs and Outputs 3

2.3 Product Interfaces 3

3. Customer Requirements 5

3.1 Replacement of RC Vehicle Receiver 7

3.2 PPM Signal Channel Selection 7

3.3 PPM Signal Calculation 8

3.4 PPM Signal Communication 8

3.5 Maintain Standard Connections 9

3.6 Secure Module Mounting 9

3.7 Maintain Vehicle Stability 10

3.8 Operation Instructions 10

3.9 System Scale 10

3.10 System Power 11

3.11 Obstacle Avoidance and Navigation Startup 12

3.12 Input GPS Destination 12
3.13 RC Vehicle Trajectory 13
3.14 Start to Finish Travel 13
3.15 RC Vehicle Trajectory Corrections 14
3.16 Obstacle Avoidance 14
3.17 Obstacle Detection 15
3.18 Object-Detection Sensors 15
3.19 Path-finding Algorithm 16
3.20 System Navigation 16
3.21 Wireless Communication 16
3.22 Location Information Transmission 17
3.23 Local Object Information Transmission 18
3.24 Live Video Feed Transmission 18
3.25 Information Reporting 19
3.26 Location Information 19
3.27 Host Application Local Obstacle Display 20
3.28 Configuration File 20
4. Packaging Requirements 21
4.1 System Will Be Fully Assembled 21
4.2 Software CD Containing User Program Included 21
4.3 Installation Instructions Included 22
5. Performance Requirements 23
5.1 Maximum RC Vehicle Speed 23
5.2 Battery Life of 20 Minutes 23
5.3 Solid Ground Movement 24
5.4 Obstacle Detection Time 5 Times/Second 24

5.5 GUI Update Time 1 Update/Second 25
5.6 GPS Response Time Less Than 3 Seconds 25
6. Safety Requirements 26
6.1 Maximum RC Vehicle Speed 26
6.2 Emergency Stop Button 26
6.3 Warning Message 27
6.4 No Sharp Edges 27
6.5 Fuse Protection 27
6.6 Connection Time-Out Control 27
7. Maintenance and Support Requirements 29
7.1 Maintenance Manual 29
7.2 End-user Manual 30
7.3 Easily Readable and Well-Commented Source Code 30
7.4 Team Autono-Mo Support 31
8. Other Requirements 32
8.1 Using English as Standard Language 32
9. Acceptance Criteria 33
9.1 Replace Vehicle Receiver. 33
9.2 Secure Mounting. 33
9.3 Standard Connections 33
9.4 Correct PPM 34
9.5 Navigate to GPS Coordinates. 34
9.6 Wireless Interraction. 34
10. Use Cases 36
10.1 Input GPS Destination 36
10.2 Uploading Configuration File 36

10.3 Emergency Stop 37
10.4 Debug Mode 37
10.5 Use Case Diagram 38
11. Feasibility Assessment 39
11.1 Scope Analysis 39
11.2 Research 40
11.3 Technical Analysis 40
11.4 Cost Analysis 41
11.5 Resource Analysis 42
11.6 Schedule Analysis 42
12. Future Items 46

Document Revision History

Revision Number	Revision Date	Description	Rationale
1.0	3/18/2012	Rough Draft of SRS	Includes rough draft of all specifications
2.0	4/22/2012	Final Draft of SRS	Includes changes from gate review feedback

List of Figures

Figure # Title Page #

Figure 1.1: Product Concept Diagram 1

Figure 2-1: Jacobia Welcome Screen 4

Figure 2.2: Jacobia Vehicle Selection Screen 4

Figure 2.3: Jacobia Mounting Instructions Screen 4

Figure 2.4: Jacobia Establish Connection Screen 5

Figure 2.5: Jacobia Safety Instructions Screen 5

Figure 2.6: Jacobia Check Connections Screen 5

Figure 2.7: Jacobia In Transit Display Screen 6

List of Tables

Table # Title Page #

Table 10.1: Expanded Use Case Table for Input GPS Destination. 36

Table 10.2: Expanded Use Case Table for Uploading Configuration File 36

Table 10.3: Expanded Use Case Table for Emergency Stop 37

Table 10.4: Expanded Use Case Table for Debug Mode 37

Table 11.1: Cost Analysis Overview 41

Table 11.2: This table shows all ILFs, EIFs, EIs, EOs, and EQs for the project. 43

Table 11.3: This table shows the Unadjusted Function Points for our system. 43

Table 11.4: This chart shows how we calculated our TDI. 44

1. Product Concept

The system will consist of a generic autonomous control module along with a host application software program. The intent of this section is to fully describe how these two components, in conjunction with one another, will make up a product. The information covered in this section will include the purpose, use, and intended audience of the aforementioned product.

1.1 Purpose and Use

The module shall provide an existing remote control(RC) vehicle with the capabilities to operate autonomously. The RC vehicle targeted for the Jacobia module is of a specific scale size, but other scales may also be able to fit the module. The module shall be able to receive GPS coordinates from a user and make the RC vehicle travel from its current GPS location to the coordinates given. While travelling to the new location, the module shall be able to detect local objects and avoid them as necessary. This module could be used for purposes such as autonomous vehicle competitions, exploration, or any other autonomous purpose. A graphical representation of the product concept can be seen in Figure 1.1 below.

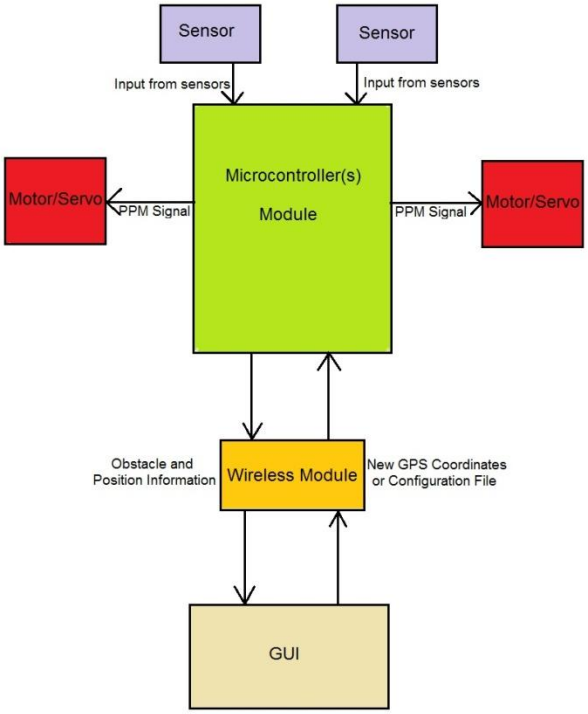


Figure 1.1 – Product Concept Diagram

1.2 Intended Audience

The intended audience would be any hobbyist interested in converting their RC vehicle into an autonomous vehicle. This would include anyone interested in learning about and implementing autonomous vehicle design.

2. Product Description and Functional Overview

The product will be a system which consists of an attachable control module along with host application software. The control module will be able to fit specified RC vehicles and provide the RC vehicles with autonomous functionality. The user, when equipped with the host application software, will be able to send GPS coordinate information to a vehicle through the use of a wireless link between the host application and the control module. The host application will house a fully-functional, easy to use GUI which will seamlessly display system information to the user.

2.1 Features and Functions

The product, when interfaced with an RC vehicle, will allow the vehicle to become an autonomous vehicle. The RC vehicle will be linked to a host application which will be able to send the RC vehicle GPS coordinates in order for the vehicle to travel to a specific location, autonomously avoiding obstacles in its path along the way. The host application will have a rich, user-friendly graphical user interface. This will allow the user to effortlessly link with the RC vehicle. This linkage will be achieved through an embedded network connection on the control module. This will allow it to transmit/receive data with the application's host computer. In order for the RC vehicle to be able to circumvent obstacles in its path, the product will make use of an efficient path finding algorithm. Referring to Figure 1.1, the product concept diagram, the linkage is obtained through the wireless technology. Furthermore, the algorithms used as well as any other instructions used will be encoded onto microchips which will be embedded within the control module portion of the product. All of these modules in conjunction with one another shall provide a seamless implementation of RC functionality and GPS navigation.

2.2 External Inputs and Outputs

The product, due to the large range of RC vehicles, expects to receive a configuration file from the host application which will configure its algorithms and instructions for peak performance for the particular type of RC vehicle that the user wishes to interface with. The user shall send this through the use of a user-friendly graphical user interface. The outputs of the system will be data corresponding to the journey of the RC vehicle. That information will include route taken, obstacles encountered, the results of the encounters, and the success of the trip.

2.3 Product Interfaces

The system will consist of two components which include a host application and a control module which will attach to the RC vehicle. The connection of these two components will be the only instance of product interface throughout the system. This interface will be accomplished through the use of wireless technology. The host application will support a rich GUI with screens similar to these:

Figure 2.1: Jacobia Welcome Screen

Figure 2.2: Jacobia RC Vehicle Selection Screen

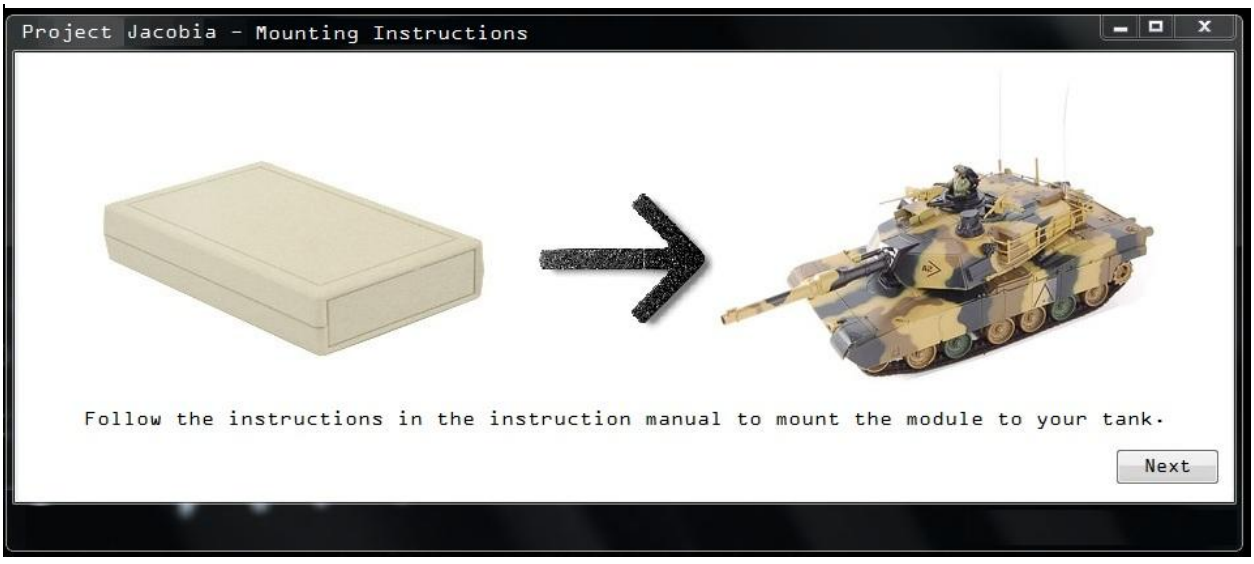

Project Jacobia - Mounting Instructions
Follow the instructions in the instruction manual to mount the module to your tank.
Next

Figure 2.3: Jacobia Mounting Instructions Screen

Figure 2.4: Jacobia Establish Connection Screen

Project Jacobia - Safety Instructions
Please read all safety instructions included in the manual before continuing.
Next

Figure 2.5: Jacobia Safety Instructions Screen

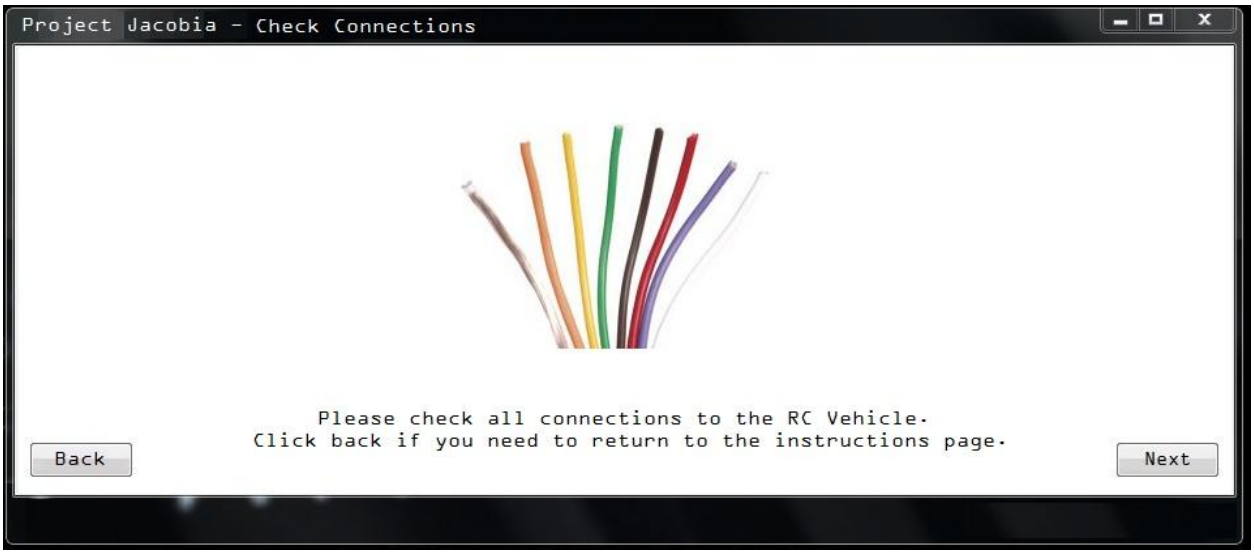

Project Jacobia - Check Connections
Please check all connections to the RC Vehicle.
Click back if you need to return to the instructions page.
Back
Next

Figure 2.6: Jacobia Check Connections Screen

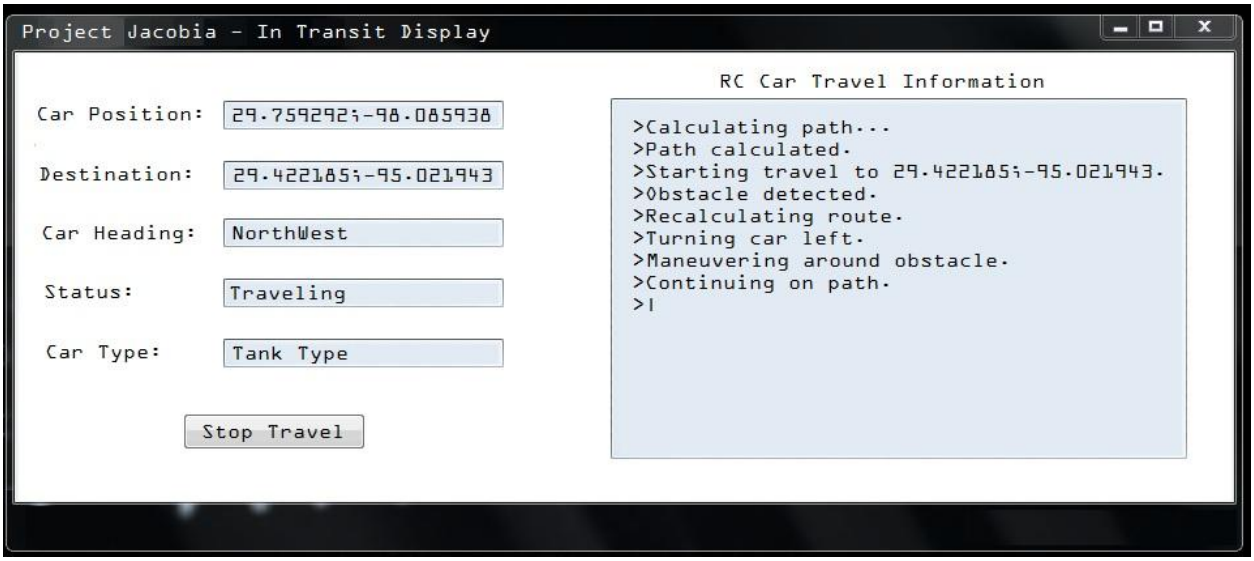


Figure 2.7: Jacobia In Transit Display Screen

3. Customer Requirements

The system shall contain all of the requirements requested by the customer for Jacobia. These are the most fundamental requirements of the system, and without meeting these, the customer shall most likely not be fully satisfied with the end product. These requirements will be closely monitored throughout the development process and major decisions will be made based on them. These requirements include functionality, use, appearance, and general attributes of the product.

3.1 Replacement of RC Vehicle Receiver

3.1.1 Description: The system shall replace the RC vehicle's receiver as a communication device for the vehicle. An RC vehicle's initial method of communication is the receiver which communicates with the transmitter (remote control), this receiver will be removed and replaced by connections to the control module.

3.1.2 Source: Chris McMurrough

3.1.3 Constraints: The receiver may be difficult to remove.

3.1.4 Standards: The control module must seamlessly act as the receiver, mimicking every bit of the original receiver's functionality.

3.1.5 Priority: 1 – Critical

3.2 Pulse Position Modulation(PPM) Signal Channel Selection

3.2.1 Description: The system shall choose appropriate channels to send PPM signals to. The module will be connected to several motors and servos which drive the vehicle. Each motor or servo will be on a separate channel. The control module will need to make decisions on which channel to send the PPM signals.

3.2.2 Source: Chris McMurrough

3.2.3 Constraints: The specific RC vehicle used must be receptive to the PPM signals sent.

3.2.4 Standards: The control module must supply the appropriate PPM signals as fast as the original architecture of the RC vehicle performed the signal appropriations.

3.2.5 Priority: 1 – Critical

3.3 PPM Signal Calculation

3.3.1 Description: The system shall calculate the correct PPM signal to send to each of the vehicle's motors and servos. After the module makes a new trajectory calculation, it will then need to calculate the correct PPM signals to send to the motors and servos.

3.3.2 Source: Chris McMurrough

3.3.3 Constraints: The PPM signal may be difficult to keep within the respective bounds of the channel receiving the transmission.

3.3.4 Standards: The signals sent must adhere to the laws of PPM which include clock synchronization between receiver and transmitter as well as a later determined maximum amount of bits that can be sent per second.

3.3.5 Priority: 1 – Critical

3.4 PPM Signal Communication

3.4.1 Description: The system shall send the correct PPM signal to the vehicle's motors and servos. After the control module calculates the correct PPM signal and decides to which channel the signal must go, it must then send the signal.

3.4.2 Source: Chris McMurrough

3.4.3 Constraints: There is a possibility of a small amount of PPM signal loss in the transmission.

3.4.4 Standards: The signals sent must adhere to the laws of PPM which include clock synchronization between receiver and transmitter as well as a later determined maximum amount of bits that can be sent per second.

3.4.5 Priority: 1 – Critical

3.5 Maintain Existing RC Vehicle Connections

3.5.1 Description: The system shall use the existing connections to the vehicle's motors and servos. Using these connectors on the control module will ensure the best compatibility. The control module will be easily maintainable if the original architecture is preserved.

3.5.2 Source: Chris McMurrough

3.5.3 Constraints: The standard connections must not be damaged during installation of the control module.

3.5.4 Standards: N/A

3.5.5 Priority: 1 – Critical

3.6 Secure Module Mounting

3.6.1 Description: The system shall mount securely to the vehicle's frame or chassis. The control module should be able to mount securely to the vehicle without the possibility of falling off. This will be accomplished by designing a mount to the vehicle's frame or chassis.

3.6.2 Source: Bill Butts

3.6.3 Constraints: The car must be able to bear the weight of the control module.

3.6.4 Standards: N/A

3.6.5 Priority: 1 – Critical

3.7 Maintain Vehicle Stability

3.7.1 Description: The system shall not cause the vehicle to become unstable. If the control module is too bulky or is mounted incorrectly then the vehicle could become unstable.

3.7.2 Source: Bill Butts

3.7.3 Constraints: N/A

3.7.4 Standards: The RC vehicle will be of $1/10^{th}$, $1/12^{th}$ or $1/16^{th}$ scale.

3.7.5 Priority: 1 – Critical

3.8 Operation Instructions

3.8.1 Description: The system shall provide the user with operation instructions. The host application shall present instructions to the user in regards to installation and use of the system itself.

3.8.2 Source: Darius Salemizadeh

3.8.3 Constraints: N/A

3.8.4 Standards: N/A

3.8.5 Priority: 1 – Critical

3.9 System Scale

3.9.1 Description: The system shall be of $1/12^{th}$ scale. The system shall be designed to support only RC vehicles which are of $1/12^{th}$ scale. The system may still be applicable for systems of $1/10^{th}$ and $1/16^{th}$ scale.

3.9.2 Source: Darius Salemizadeh

3.9.3 Constraints: Vehicle design may prevent the system from operating on all vehicles of that scale.

3.9.4 Standards: The system shall be based upon $1/10^{th}$, $1/12^{th}$, or $1/16^{th}$ scale RC vehicle properties.

3.9.5 Priority: 2 – High

3.10 System Power

3.10.1 Description: The system shall utilize the existing power supply of the RC vehicle in order to supply the power for the system's operations.

3.10.2 Source: Chris McMurrough

3.10.3 Constraints: Since the battery being used is the existing battery of the RC vehicle, the performance will depend on the drains on the battery from other components of the vehicle.

3.10.4 Standards: NA

3.10.5 Priority: 1 – Critical

3.11 Obstacle Avoidance and Navigation Startup

3.11.1 Description: The system shall contain software that initializes the Obstacle Avoidance and Navigation System. The system shall contain software that allows the system to calculate where it currently is, what direction it is facing, and several other start-up checks that need to be run before the RC vehicle starts its journey. The system shall always perform the same checks every time it starts up.

3.11.2 Source: Chris McMurrough

3.11.3 Constraints: N/A

3.11.4 Standards: National Marine Electronics Association 0183-standard

3.11.5 Priority: 1 – Critical

3.12 Input GPS Destination

3.12.1 Description: The system shall allow for the end-user to input the GPS coordinates that they wish the vehicle to travel to from its current location.

3.12.2 Source: Chris McMurrough

3.12.3 Constraints: The GPS coordinates of the destination may be off by +/- 1 meter depending on the GPS unit within the vehicle.

3.12.4 Standards: National Marine Electronics Association 0183-standard.

3.12.5 Priority: 1 – Critical

3.13 RC Vehicle Trajectory

3.13.1 Description: The system shall be able to determine the RC vehicle's trajectory through the use of on-board devices.

3.13.2 Source: Chris McMurrough

3.13.3 Constraints: The system may not be able to provide real-time calculations and may be just near-real-time.

3.13.4 Standards: National Marine Electronics Association 0183-standard

3.13.5 Priority: 1 – Critical

3.14 Start to Finish Travel

3.14.1 Description: The system shall be able to make an RC vehicle travel from one set of GPS coordinates to the end-user specified destination coordinates. The system shall travel along a created path from its current location to the destination location input by the end-user.

3.14.2 Source: Chris McMurrough

3.14.3 Constraints: The system may not be able to travel to the specified destination coordinates using its original path if it is in a higher or lower location or if there is an obstacle on its path.

3.14.4 Standards: National Marine Electronics Association 0183-standard

3.14.5 Priority: 1 – Critical

3.15 RC Vehicle Trajectory Corrections

3.15.1 Description: The system shall include metrics that test whether or not the RC vehicle is actually going on the right path, and if it is not, it will correct the RC vehicle's trajectory and put it back on the right path.

3.15.2 Source: Chris McMurrough

3.15.3 Constraints: Corrections may be limited by the quality of the hardware used in the system.

3.15.4 Standards: N/A

3.15.5 Priority: 1 – Critical

3.16 Obstacle Avoidance

3.16.1 Description: The system shall control the RC vehicle in such a way as to avoid obstacles along its path. The system shall use sensory input to detect obstacles and will then correct its path to avoid such obstacles to ensure a successful journey.

3.16.2 Source: Chris McMurrough

3.16.3 Constraints: The system may not be able to detect certain obstacles such as those that are too small to detect.

3.16.4 Standards: N/A

3.16.5 Priority: 1 – Critical

3.17 Obstacle Detection

3.17.1 Description: The system shall have sensors that can detect potential obstacles. The system shall utilize sensors to view the environment around the RC vehicle, detecting anything that may cause problems along the path created so that adjustments can be made.

3.17.2 Source: Chris McMurrough

3.17.3 Constraints: The sensors may not be sensitive enough to be able to detect all potential obstacles.

3.17.4 Standards: N/A

3.17.5 Priority: 1 – Critical

3.18 Object-Detection Sensors

3.18.1 Description: The system shall have sensors that can detect the distance to and the relative size of local obstacles. The system shall have sensors that can detect the distance between the RC vehicle and the obstacles detected so that the system can accurately correct its path.

3.18.2 Source: Chris McMurrough

3.18.3 Constraints: The sensors may not be of a high enough quality to be able to detect all potential obstacles.

3.18.4 Standards: N/A

3.18.5 Priority: 1 – Critical

3.19 Path-finding Algorithm

3.19.1 Description: The system shall use an efficient path-finding algorithm that will find a quick and safe route to the destination GPS coordinates. This algorithm will take into account obstacles found with the on-board sensors.

3.19.2 Source: Chris McMurrough

3.19.3 Constraints: The system may not be able to create a path depending on the environment, as well as the quality of the sensors used.

3.19.4 Standards: N/A

3.19.5 Priority: 1 – Critical

3.20 System Navigation

3.20.1 Description: The control module shall make use of a weighted system when calculating its heading path that will take into account servo odometry and GPS signaling.

3.20.2 Source: Chris McMurrough

3.20.3 Constraints: Bad readings from GPS signaling will create a need for the system to shift the weighting assigned to the two components of the navigation calculation.

3.20.4 Standards: N/A

3.20.5 Priority: 2 – High

3.21 Wireless Communication

3.21.1 Description: The system shall interface wirelessly with the host application. Any input or output from the system to the host application will be done using wireless technology.

3.21.2 Source: Chris McMurrough

3.21.3 Constraints: The system should be in range of the wireless technology and within a maximum range of 120 feet. The system uses a local wireless network therefore the LAN must be online and operational for the system to connect.

3.21.4 Standards: The system shall follow the wireless communications standard of IEEE 802.15.4.

3.21.5 Priority: 2 – High

3.22 Location Information Transmission

3.22.1 Description: The system, when traveling from one location to another, shall send relative location information to the host application periodically.

3.22.2 Source: Chris McMurrough

3.22.3 Constraints: The system should be in range of the wireless technology and within a maximum range of 120 feet. The system uses a local wireless network therefore the LAN must be online and operational for the system to connect.

3.22.4 Standards: The system shall follow the wireless communications standard of IEEE 802.15.4.

3.22.5 Priority: 2 – High

3.23 Local Object Information Transmission

3.23.1 Description: The system shall send local obstacle information to the host application. During the transit state of the system, the system shall send information related to local obstacles encountered by the system.

3.23.2 Source: Chris McMurrough

3.23.3 Constraints: The system should be in range of the wireless technology and within a maximum range of 120 feet. The system uses a local wireless network therefore the LAN must be online and operational for the system to connect.

3.23.4 Standards: The system shall follow the wireless communications standard of IEEE 802.15.4.

3.23.5 Priority: 2 – High

3.24 Live Video Feed Transmission

3.24.1 Description: The system shall capture and send a live video feed of its surroundings to the host application which will then display the video feed through the GUI to the user. The video feed will capture things such as local obstacles, terrain, and overall journey taken by the RC vehicle.

3.24.2 Source: Chris McMurrough

3.24.3 Constraints: The system should be in range of the wireless technology and within a maximum range of 120 feet. The system uses a local wireless network therefore the LAN must be online and operational for the system to connect. Also, there may be some latency in the transmission of the video feed.

3.24.4 Standards: The system shall follow the wireless communications standard of IEEE 802.15.4.

3.24.5 Priority: 5 – Future

3.25 Information Reporting

3.25.1 Description: The system shall include a host application that will be installed on the user's computer and will have a GUI that will display the current state of the system. The different types of states include stopped and in transit, which both only refer to the current movement of the RC vehicle.

3.25.2 Source: Chris McMurrough

3.25.3 Constraints: The information displayed may have a small amount of lag due to the quality of the system's wireless communication devices. The size of the user's screen will serve as a constraint to the amount of information the application can show on its GUI.

3.25.4 Standards: N/A

3.25.5 Priority: 3 – Moderate

3.26 Location Information

3.26.1 Description: The host application shall show the location information of the system. The subsystem which is an application on the laptop, shall display the current location and other information related to the location i.e. GPS coordinates, Geo-location, maps etc.

3.26.2 Source: Chris McMurrough

3.26.3 Constraints: The information displayed may have a small amount of lag due to the quality of the system's wireless communication devices. The size of the user's screen will serve as a constraint to the amount of information the application can show on its GUI.

3.26.4 Standards: N/A

3.26.5 Priority: 3 – Moderate

3.27 Host Application Local Obstacle Display

3.27.1 Description: The host application shall show the location of local obstacles encountered by the system. The host application shall display the location of the current obstacles encountered by the system in the host application GUI.

3.27.2 Source: Chris McMurrough

3.27.3 Constraints: The information displayed may have a small amount of lag due to the quality of the system's wireless communication devices. The size of the user's screen will serve as a constraint to the amount of information the application can show on its GUI.

3.27.4 Standards: N/A

3.27.5 Priority: 3 – Moderate

3.28 Configuration File

3.28.1 Description: The host application shall upload the configuration file to the system in order to operate within the bounds of the chosen RC vehicle. This configuration file will include all the pertinent information needed to configure to a certain RC vehicle type.

3.28.2 Source: Chris McMurrough

3.28.3 Constraints: N/A

3.28.4 Standards: N/A

3.28.5 Priority: 5 – Future

4. Packaging Requirements

Every large scale product implementation team must pay careful attention to how its product is presented to the intended customer. Team Autono-Mo's product is no different. Therefore, this section exists in order to describe the look of the product upon delivery as well as the components that make up the system.

4.1 System Will Be Fully Assembled

4.1.1 Description: The control module and mount will be delivered as one complete piece. Only mounting and connecting will be required.

4.1.2 Source: Team Autono-Mo

4.1.3 Constraints: Physical strength of the assembly.

4.1.4 Standards: The system will be created for an RC vehicle of $1/12^{th}$ scale and the system may also be able to fit those of $1/10^{th}$ and $1/16^{th}$ scale.

4.1.5 Priority: 2 – High

4.2 Software CD Containing User Program Included

4.2.1 Description: The host application will be put on a CD and delivered with the module. The CD will allow the user to install the host application on their computer.

4.2.2 Source: Team Autono-Mo

4.2.3 Constraints: The host application may not be cross-platform compatible. Only Microsoft Windows 7 will be supported.

4.2.4 Standards: 700MB Compact Disk and software compliant with Microsoft Windows 7 platform specifications.

4.2.5 Priority: 2 – High

4.3 Installation Instructions Included

4.2.1 Description: The installation of the control module to the vehicle will require an instruction manual. The manual will cover mounting to the vehicle, vehicle type supported, and user instructions for the software and configuration file.

4.2.2 Source: Team Autono-Mo

4.2.3 Constraints: N/A

4.2.4 Standards: N/A

4.2.5 Priority: 2 - High

5. Performance Requirements

This section describes performance requirements that the module will adhere to during operation. This includes items such as: speed limitations, battery limitations, sensory input response times, and GUI update times. These are all important factors in ensuring the overall success of Jacobia.

5.1 Maximum RC Vehicle Speed

5.1.1 Description: The vehicle shall not travel any faster than 10 miles per hour in order to allow sensors to take accurate measurements. Jacobia will limit the vehicle's speed to this maximum speed limit.

5.1.2 Source: Team Autono-Mo

5.1.3 Constraints: N/A

5.1.4 Standards: N/A

5.1.5 Priority: 2 – High

5.2 Battery Life of 20 Minutes

5.2.1 Description: The vehicle must be able to travel for at least 20 minutes on a single battery charge.

5.2.2 Source: Team Autono-Mo

5.2.3 Constraints: Since the battery being used is the existing battery of the RC vehicle, the performance will depend on the drains on the battery from other components of the vehicle.

.**5.2.4 Standards:** N/A

5.2.5 Priority: 1 - Critical

5.3 Solid Ground Movement

5.3.1 Description: The vehicle will only be designed to operate on solid ground. Mud, water, extreme inclines, and other unfavorable landscapes will not be covered by the vehicle's capabilities.

5.3.2 Source: Team Autono-Mo

5.3.3 Constraints: The vehicle will not be able to determine its terrain. This will be determined by the operator.

5.3.4 Standards: N/A

5.3.5 Priority: 1 – Critical

5.4 Obstacle Detection Time 5 Times/Second

5.4.1 Description: The module will detect obstacles 5 times/second. This will keep the module's obstacle detection properly updated in relation to its speed.

5.4.2 Source: Team Autono-Mo

5.4.3 Constraints: Design of the specific sensors and operating code may limit this rate.

5.4.4 Standards: N/A

5.4.5 Priority: 2 - High

5.5 GUI Update Time 1 Update/Second

5.5.1 Description: The GUI will need to be updated continuously in time intervals to ensure that the user has the most current information about the vehicle. One update/second should be fast enough to update the user.

5.5.2 Source: Team Autono-Mo

5.5.3 Constraints: The data transfer rate for the wireless module may limit this rate.

5.5.4 Standards: The system shall follow the wireless communications standard of IEEE 802.15.4.

5.5.5 Priority: 2 – High

5.6 GPS Response Time Less Than 3 Seconds

5.6.1 Description: The module will need to respond to new user GPS coordinates within a timely manner. Less than 3 seconds will be enough time to change the vehicles course from its current position.

5.6.2 Source: Team Autono-Mo

5.6.3 Constraints: The data rate of the wireless technology in conjunction with the response time of the control module to the incoming GPS coordinates.

5.6.4 Standards: The system shall follow the wireless communications standard of IEEE 802.15.4 as well as the GPS standard of the National Marine Electronics Association 0183.

5.6.5 Priority: 3 - Moderate

6. Safety Requirements

In order to assure the safety and well-being of every user when operating Team Autono-Mo's system, many safety requirements have been engineered. These requirements serve as safeguards against injury for the potential users operating the system.

6.1 Maximum RC Vehicle Speed

6.1.1 Description: The system shall support an RC vehicle with a top speed of no more than 10 miles per hour. This will allow the system to be much less dangerous if operating near people.

6.1.2 Source: Chris McMurrough

6.1.3 Constraints: The RC Vehicle used may not be of a high enough quality for us to accurately limit maximum speed to 10 miles per hour.

6.1.4 Standards: N/A

6.1.5 Priority: 1 – Critical

6.2 Emergency Stop Button

6.2.1 Description: The system shall furnish an emergency stop button. The host application shall house an emergency stop button which, when pressed, will remotely stop the automated RC vehicle's movement. The RC vehicle shall stop within 2 seconds of pressing the button.

6.2.2 Source: Chris McMurrough

6.2.3 Constraints: The design of the system will limit the reaction time of the button.

6.2.4 Standards: N/A

6.2.5 Priority: 1 – Critical

6.3 Warning Message

6.3.1 Description: The system shall display a warning message prior to use. The system shall project a safety message, through the host application, to the user describing the developer intent of the product in regards to RC vehicle speed, size, and other pertinent items. The system shall use an eye catching format for the warning message.

6.3.2 Source: Chris McMurrough

6.3.3 Constraints: N/A

6.3.4 Standards: N/A

6.3.5 Priority: 1 – Critical

6.4 No Sharp Edges

6.4.1 Description: The system shall be designed in such a way as to not contain sharp edges. This will allow the module to not cause harm to anyone that it could come into contact with.

6.4.2 Source: Chris McMurrough

6.4.3 Constraints: N/A

6.4.4 Standards: N/A

6.4.5 Priority: 1 – Critical

6.5 Fuse Protection

6.5.1 Description: The system shall contain electronic components that will all be fuse protected. This will allow the system to not electrocute or shock anyone that comes into contact with the module while powered on.

6.5.2 Source: Chris McMurrough

6.5.3 Constraints: N/A

6.5.4 Standards: N/A

6.5.5 Priority: 1 – Critical

6.6 Connection Time-Out Control

6.6.1 Description: The control module, when disconnected from the host application, shall adhere to a maximum timeout period of 10 seconds. After this 10 second timeout period has been reached, the system shall halt all operations and the vehicle shall stop moving.

6.6.2 Source: Chris McMurrough

6.6.3 Constraints: N/A

6.6.4 Standards: N/A

6.6.5 Priority: 2 – High

7. Maintenance and Support Requirements

The system shall have a limited expected lifetime due to continual use or wear and tear. The battery life of the system shall also be limited. Due to these constraints, the system shall eventually require maintenance at one point or another. Since this system comprises of an attachable module, a wireless communications device and a software interface, many of the maintenance issues will be with batteries, cables, antennas, sensors, and software. The following requirements have to be met in order to make maintenance as simple as possible for the end-user.

7.1 Maintenance Manual

7.1.1 Description: The system shall be delivered with a maintenance manual. The system shall include a maintenance manual that tells the end-user how to replace parts and/or fix common issues with the system, such as: exchanging batteries, exchanging parts, cleaning the system, and other simple tasks.

7.1.2 Source: Team Autono-Mo

7.1.3 Constraints: The maintenance manual shall include a list of all parts within the system, and how to replace those that are replaceable. Some parts of the system will not be easily replaceable, and in such a case, the end-user should contact Team Autono-Mo.

7.1.4 Standards: N/A

7.1.5 Priority: 3 – Moderate

7.2 End-user Manual

7.1.1 Description: The system shall be delivered with an end-user's manual. The system shall include an end-user's manual to allow the end-user to fully operate and understand the complete system. This includes (but is not limited to) tasks such as: installation and operation of included software, attaching module to RC vehicle, configuring the system for the specific vehicle, starting the system, explanations of the transit screen, and other pertinent tasks.

7.2.2 Source: Team Autono-Mo

7.2.3 Constraints: N/A

7.2.4 Standards: N/A

7.2.5 Priority: 2 – High

7.3 Easily Readable and Well-Commented Source Code

7.3.1 Description: The system shall have easily readable and well-commented source code. The system shall include source code that is easy to understand and can be comprehended through the use of comments and meaningful code. The end-user should be able to easily read through the source code to find what they are looking for.

7.3.2 Source: Team Autono-Mo

7.3.3 Constraints: N/A

7.3.4 Standards: N/A

7.3.5 Priority: 2 – High

7.4 Team Autono-Mo Support

7.4.1 Description: Team Autono-Mo shall be available to fix software and hardware issues. Should the system require maintenance that the end-user cannot do through the use of the Maintenance Manual, Team Autono-Mo should be contacted in order to fix the issue(s). This service will be provided free of charge.

7.4.2 Source: Team Autono-Mo

7.4.3 Constraints: Team Autono-Mo will not be responsible for maintenance issues after August, 2012.

7.4.4 Standards: N/A

7.4.5 Priority: 1 – Critical

8. Other Requirements

This section defines requirements that are not defined in any other section of this document.

8.1 Using English as a Standard Language

8.1.1 Description: Jacobia will use English as its official language throughout its implementation.

8.1.2 Source: Team Autono-Mo

8.1.3 Constraints: N/A.

8.1.4 Standards: N/A.

8.1.5 Priority: 1 – Critical

9. Acceptance Criteria

Project Jacobia must satisfy the customer requirements in order to be labeled acceptable. This section outlines the criteria in which the module will be accepted by the customer. The acceptance criteria are derived from specific customer requirements defined in section 3.

9.1 Verify that the module can effectively replace the vehicle's receiver.

9.1.1 Requirement(s) addressed

Requirement 3.1: Replacement of RC Vehicle Receiver.

9.1.2 Verification Procedure: After the module is connected, it will be placed in a debug mode. This mode will test the connections and controls of the motors and servos. If the module can accurately control the motors and servos then this criteria shall be met.

9.2 Verify that the module mounts securely to the vehicle and causes no instability upon operation.

9.2.1 Requirement(s) addressed:

Requirement 3.7: Secure Model Mounting.
Requirement 3.8: Maintain Vehicle Stability.

9.2.2 Verification Procedure: After the module is mounted and connected to the vehicle, it will run in debug mode. During this time, signs of instability in the module, mount, and vehicle operation will be observed. If there are no instabilities then the module shall meet these criteria.

9.3 Verify that the module uses only standard connections

9.3.1 Requirement(s) addressed:

Requirement 3.5: Maintain Standard Connections.

9.3.2 Verification Procedure: The connections used will be observed to be standards used throughout the field. If the connections are easily verifiable then this criterion will be met.

9.4 Verify that the module can create the correct PPM signals and send the signals to the correct channels.

9.4.1 Requirement(s) addressed:

Requirement 3.2: PPM Signal Channel Selection.
Requirement 3.3: PPM Signal Calculation.
Requirement 3.4: PPM Signal Communication.

9.4.2 Verification Procedure: The module will be put into debug mode to verify the PPM signals to the motors and servos. If the module predictably controls the motors and servos then this criterion will be met.

9.5 Verify that the module can navigate the vehicle from its current location to a user defined GPS coordinate.

9.5.1 Requirement(s) addressed:

Requirement 3.13: Input GPS Destination.
Requirement 3.15: Start to Finish Travel.
Requirement 3.16: RC Vehicle Trajectory Corrections.
Requirement 3.20: Path-Finding Algorithm.

9.5.2 Verification Procedure:

The module will be given a GPS coordinate. It will then calculate its path to travel from its current location to the user defined location. If the vehicle successfully navigates to the new location it shall meet this criterion.

9.6 Verify that the user can interact wirelessly with the module and receive location, path and obstacle information through GUI software.

9.6.1 Requirement(s) addressed:

Requirement 3.21: Wireless Communication.
Requirement 3.22: Location Information Transmission.
Requirement 3.23: Local Object Information Transmission.
Requirement 3.24: Information Reporting.
Requirement 3.25: Location Information.
Requirement 3.26: Host Application Local Obstacle Display.
Requirement 3.27: Configuration File.

9.6.2 Verification Procedure: The user shall startup the GUI software. The user will then upload a configuration file for the vehicle being used. While the vehicle is in operation information regarding location, obstacle location and obstacle display. If all of this information is correct then this criterion will have been met.

10. Use Cases

This section defines all of the user interactions which will occur with Jacobia.

10.1 Input GPS Destination

10.1.1 Scenario: The user enters the destination that the RC vehicle should travel to in proper GPS format. This data will be analyzed by the system, and the system will then direct the car to the specified destination.

10.1.2 Actor(s): User

10.1.3 Expanded Use Case Table:

Input GPS Destination	
Actor: User	System: Jacobia
	0. System is in ready stage of operation
1. TUCBW User inputs the GPS destination into the system using the GUI.	2. System operates and travels to provided destination.
3. TUCEW User sees the vehicle has reached the final destination.	

Table 10.1: Expanded Use Case Table for Input GPS Destination.

10.2 Uploading Configuration File

10.2.1 Scenario: The user shall upload a configuration file which tells the system about the vehicle which it is controlling.

10.2.2 Actor(s): User

10.2.3 Expanded Use Case Table:

Uploading Configuration File	
Actor: Developer	System: Jacobia
System:	0. System is in online mode connected to the subsystem wirelessly.
1. TUCBW User uploads the Configuration file to the system.	2. System confirms the configuration files being uploaded.
3. TUCEW User sees confirmation message about files being uploaded.	

Table 10.2: Expanded Use Case Table for Uploading Configuration File

10.3 Emergency Stop

10.3.1 Scenario: In the event of an unknown failure or malfunction, the user will have the ability to hit the emergency stop button to disable the module.

10.3.2 Actor(s): User

10.3.3 Expanded Use Case Table:

Emergency Stop	
Actor: User	System: Jacobia
	0. System is in operation mode and is running.
1. TUCBW User presses emergency stop on the system.	2. System stops operating.
3. TUCEW User sees the system being shut down.	

Table 10.3: Expanded Use Case Table for Emergency Stop

10.4 Debug Mode

10.4.1 Scenario: Testing of the module will be done in a debug mode. This mode will allow for testing of a specific functionality.

10.4.2 Actor(s): User

10.4.3 Expanded Use Case Table:

Debug Mode	
Actor: Developer	System: Jacobia
	0. System is in operation mode and online.
1. TUCBW User choose debug mode on the system.	2. System operates on the debug mode.
3. TUCEW User sees the system in the debug mode.	

Table 10.4: Expanded Use Case Table for Debug Mode

10.5 Use Case Diagram

10.5.1 Scenario: The user shall be able to enter GPS coordinates, upload configuration files, use an emergency stop if needed, and test using a debug mode.

10.5.2 Actor(s): User

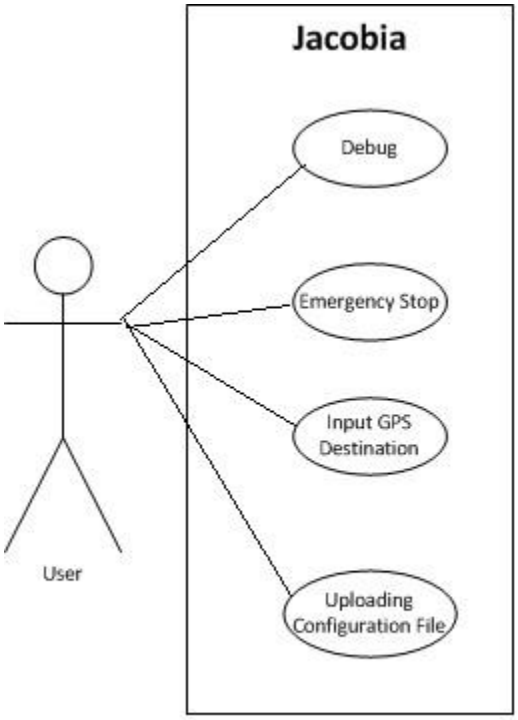


Figure 10.1: Use Case Diagram for Project Diagram

11. Feasibility Assessment

This section of the SRS is an analysis of the feasibility of project Jacobia. This includes several key features: scope analysis, research analysis, cost analysis, resource analysis, and schedule analysis. Since project Jacobia is still only in the design phase, Team Autono-Mo can only make judgments on the feasibility of the system based on preliminary research and work done towards the system. However, Team Autono-Mo feels that the system is indeed a feasible project and will be able to be completed by August, 2012.

11.1 Scope Analysis

The scope of work for our critical priority requirements is reasonable and within our capabilities. Building a prototype that allows for the autonomous control of an RC vehicle that complies with all of our requirements is feasible given our timeline and skill set. We can comfortably state this after doing some research into the field of autonomous RC vehicles. We have found several projects similar to our own, and they were able to complete their projects within a relatively similar timeframe to that of our own. We felt that implementing a path-finding algorithm would be time consuming, but we found that there are several resources online to help with this portion of our project.

With all of these factors in mind, there are several customer requirements that comprise the majority of our project. These are the requirements that will most likely take the largest amount of time to implement, and as such, we will be giving them the most attention. Those requirements are as follows:

- Customer Requirement 3.15: The system shall be able to travel from one GPS coordinate to another GPS coordinate.
- Customer Requirement 3.17: The system shall be able to avoid local obstacles along its path.
- Customer Requirement 3.4: The system shall be able to interface with standard RC vehicle PPM signals.
- Customer Requirement 3.21: The system shall be able to interface wirelessly with a host application.
- Customer Requirement 3.27: The system shall have a GUI for interfacing with the user.

11.2 Research

Through our initial research sessions, Team Auto-Mo was able to find other projects which were very similar to our own in regards to functionalities delivered. We used them as a reference for determining several different factors for our project such as time requirements, cost requirements, parts required, algorithms used, and other pertinent fields of interest.

11.2.1 Path finding algorithm

Through our research of several path finding algorithms, we have determined that this portion of our project will not be difficult to implement. There are several examples on the Internet and several of our group members are students of Dr. Huber's Robotics class, in which we are being taught how to utilize these algorithms. This portion of our project should not be insurmountable.

11.2.2 Obstacle Detection and Avoidance

The next biggest field of research for our project is related to obstacle detection and avoidance. We will be using several sensors mounted on the module in order to allow our system to 'view' the surrounding environment. Once we take in this information, the system must be able to calculate how to avoid the detected obstacles. We have done some research on the types of sensors required and how to use them, but once it comes to implementation, more research will be required in the specifics of how the sensors interact with our system.

11.2.3 RC Vehicle PPM signals

As stated previously, our system needs to be able to communicate with the servos of the RC vehicle. Thus, research has been conducted on how this communication takes place in a normal RC vehicle. Since our vehicle will bypass the current receiver system within the vehicle, we will have to conduct more research on how to implement our module in such a way as to communicate with the servos correctly.

11.3 Technical Analysis

This project requires a wide range of technical skills in the following areas: hardware design, RC vehicle integration, GUI design, and inter-device communication.

Hardware design includes everything that will be contained within the module for project Jacobia. This includes, but is not limited to: processors, a GPS unit, batteries, sensors, wireless modules, and miscellaneous circuitry. All of these components are needed to fit into some type of relatively small module that can be easily attached to an RC vehicle. These are all important components to achieving our goal of creating an autonomous RC vehicle. The largest complication in regards to these components is successfully fitting them into a small module and attaching them to each other so that they can communicate.

RC vehicle integration includes everything that is needed to let our module communicate with the servos of the RC vehicle. This will require us to send our own self-generated PPM signals to the servos so that the vehicle can effectively follow our instructions. Chris McMurrough has informed us about this process, and has said that he is willing to help us with this portion of the design. The

largest complication with this is making sure that the correct PPM signals are sent to the servos at the right times. Through experimentation, we will not have any problem getting this correct.

GUI design includes everything that is needed to allow the end-user to interface with the device attached to their RC vehicle. This GUI will have features such as current GPS position, destination GPS position, current heading, vehicle type, vehicle status, and a vehicle information text box that includes all of the decisions the module is making. The largest complication that arises with this portion of project Jacobia is creating a GUI that is easy to use and appeals to the user visually.

Inter-device communication includes everything that is needed to allow the system to intelligently decide how it will proceed to the destination GPS position while avoiding obstacles in its path. This includes processes such as calculating current GPS coordinates, calculating heading, detecting obstacles, changing course to avoid obstacles, and constantly correcting its self to reach the destination GPS position. The largest complication with this system is making all of the devices effectively communicate with each other in a real-time situation.

11.4 Cost Analysis

Since our budget is $800, we felt that it would be a good idea to research the costs of our components early in order to get an idea of how much money we would need to complete this project. After analyzing the costs of components, we believe it is feasible to create the unit, and we will most likely go under our budget limit. Below is a chart with the corresponding prices of most of our required components.

Component	Low-End Cost	High-End Cost
Sensors	$45	$200
Processors	$5	$45
RC Vehicle	$50	$300
Batteries	$13	$24
Circuit Components	$0	$50
HBridge	$3.75	$15
GPS	$40	$100
Compass w/Tilt Compensation	$29.95	$50
Wireless Module	$30	$70
Totals	**$216.70**	**$854.00**

Table 11.1: Cost Analysis Overview

Table 11.1 is a brief overview for the cost of hardware for our project. This list is by no means complete, and this range will change several times over the course of our project. There's a good chance that this list will eventually expand once we get into the stages of implementation, but we'll

have enough of our budget left to account for such additions. Overall, we feel that we can confidently consider this project feasible with our given budget. As a side note, this estimate does not include taxes or shipping costs. We may also be able to get some of these things donated to us or salvage them from other projects. We also allocated a maximum amount $300 for our RC vehicle, but we will probably not spend that much. In a worst case scenario, we would go over our budget by $54 with this estimate, but this is very unlikely and easily mitigated.

11.5 Resource Analysis

Team Autono-Mo consists of one Software Engineering student, one Computer Science student, and two Computer Engineering students. We feel that our wide range of skills will allow us to effectively carry out this project, but there are certain limitations we face. We estimate that the most difficult task will be interfacing all of the components together once we get into the stages of implementation that require all of our systems to communicate. The hardware portion of this integration is going to be the most difficult thing for us. Despite having two Computer Engineers on our team, they are both somewhat lacking in experience with combining the amount of hardware that will be required for our system. Because of this, we will be asking our sponsor, Chris McMurrough to help us with the main parts of the hardware integration. Doing something incorrect and potentially damaging a piece of hardware is unacceptable, so we will do everything in our power to prevent this.

The majority the project encompasses work that at least one or more members of Team Autono-Mo are familiar with. Each member has had several years of experience in programming, the two Computer Engineering members have had lots of experience with reading input data and using that data to make decisions within a system, our Computer Science and Software Engineering members have both had experience with creating a GUI system and implementing algorithms, and the other skills required of us will be learned as we further develop this project. We have all had lots of experience with learning new languages and systems within relatively small timeframes, so this will come easily for the team during the course of the project as well. From a technical standpoint, Project Jacobia is a very feasible endeavor for Team Autono-Mo.

11.6 Schedule Analysis

Team Autono-Mo utilized three methods of time estimation and formulated a feasible schedule based on these estimations. Through the use of Function Points, Use-Cases and COCOMO, we were able to formulate an average estimate of 5.13 months, which is within our scheduled time-frame. Because of this, we feel that our schedule is feasible.

Function Points
The primary method used to estimate the size of Project Jacobia was that of function point calculation. The tables below show this estimation.

Function Name	Function Type	Record Element Type	Data Element Type	File Types Referenced	Unadjusted FPs
Receiver to PPM	ILF	5-7(channels)	8	1-2	10
Microcontroller	ILF	n/a	n/a	1-2	10
Navigational data(GPS)	EI/EO	1	n/a	1	4
Sensor data	EI/EO	2-4	n/a	1-2	4
Image Processing data	EI/EO	1-2	n/a	1	3
Web Service	EI/EO	1	1	1	2
Initial data (from computer)	EI	1	n/a	1-2	5
Data base	EQ	n/a	n/a	1	1
Other hardware related data	EIF	n/a	n/a	n/a	3

Table 11.2 This table shows all ILFs, EIFs, EIs, EOs, and EQs for the project.

Function Type	Complexity	Multiplier	Line Item Sub-Total	Section Total
ILF	0 Low	x 7 =	0	
	20 Average	x 10 =	200	
	0 High	x 15 =	0	200
EIF	0 Low	x 5 =	0	
	10 Average	x 7 =	70	
	0 High	x 10 =	0	70
EI	2 Low	x 3 =	6	
	12 Average	x 4 =	48	
	3 High	x 0 =	0	54
EQ	0 Low	x 3 =	0	
	1 Average	x 4 =	4	
	8 High	x 6 =	48	52
EO	0 Low	x 4 =	0	
	6 Average	x 5 =	30	
	0 High	x 7 =	0	30
			Unadjusted Function Point Count:	406

Table 11.3 This table shows the Unadjusted Function Points for our system.

After this was complete, we then calculated the Value Adjusted Factor based on the Total Degree of Influence chart below.

14 factors	Points (0-5)
1. Data Communications	5
2. Distributed Data Processing	5
3. Performance	4
4. Heavily Used Configuration	5
5. Transaction Rate	4
6. On-line Data Entry	1
7. End -User Efficiency	4
8. On-line Update	5
9. Complex Processing	5
10. Reusability	5
11. Installation Ease	4
12. Operational Ease	5
13. Multiple Sites	1
14. Facilitate Change	5
Total Degree of Influence(TDI)	53

Table 11.4 This chart shows how we calculated our TDI.

After this, we then calculated our Adjusted FP Count. This will let us know our final value for our function points.

VAF = (TDI*0.01) + 0.65
(Value Adjusted Factor)
= 53 * 0.01 + 0.65
= 1.18
Adjusted FP Count = Unadjusted FP Count * VAF = 406 * 1.18
Adjusted FP Count =479

This total gives us an estimation of 4.4 months to complete our project.

Use-case Points
Besides using function point estimation, we also used the method of use-case points.

Previous projects stated that there was 40 FP/Use-case, and 10 use cases associated with their system. If we use this same estimation, we get our estimated total hours to be:
10 * 40 = 400 total hours

This total gives us an estimation of 4 months to complete our project.

COCOMO

We also utilized COCOMO to estimate how long our project would take, and we were able to come up with a feasible estimate. Below are our calculations:

Effort Applied (E) = $a_b(SLOC)^{b}_{b}$ **[man-months]**
Development Time (D) = $c_b(\text{Effort Applied})^{d}_{b}$ **[months]**
People required (P) = Effort Applied / Development Time **[count]**

Effort Applied (E) = $3.6\ (5k)^{1.2}$
= 3.6 * 6.89
= 24.83 (mans-months)

Development time (D) = $2.5 * 24.83^{0.32}$
= 6.98 (months)

When trying to discern the feasibility of our project in terms of time allotted to complete the task, using the above time estimation techniques, we are very confident that we will be able to complete the project in the required amount of time.

12. Future Items

This section consists of items that Team Autono-Mo has deemed noncritical to the initial system design. These items are features that would be implemented in future builds of the project and seamlessly integrated with the original system design.

12. 1 Customer Requirement 3.23: Live Video Feed Transmission

12.1.1 Requirement Description: The system shall capture and send a live video feed of its surroundings to the host application which will then display the video feed through the GUI to the user. The video feed will capture things such as local obstacles, terrain, and overall journey taken by the RC vehicle.

12.1.2 Constraint: Time: The time required to implement a non-critical requirement may not be available to the team due to the abundance of critical requirements that must be met in the allotted time frame.

12. 2 Customer Requirement 3.8: Operation Instructions

12.2.1 Requirement Description: The system shall provide the user with operation instructions. The host application shall present instructions to the user in regards to installation and use of the system itself.

12.2.2 Constraint: Time: The time required to implement a non-critical requirement may not be available to the team due to the abundance of critical requirements that must be met in the allotted time frame.

12. 3 Customer Requirement 3.27: Configuration File

12.3.1 Requirement Description: The host application shall upload the configuration file to the system in order to operate within the bounds of the chosen RC vehicle. This configuration file will include all the pertinent information needed to configure to a certain RC vehicle type.

12.3.2 Constraint: Time: The time required to implement a non-critical requirement may not be available to the team due to the abundance of critical requirements that must be met in the allotted time frame.